



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

SYÖPÄTUTKIMUSTIETOKANNAN KÄYTTÖSOVELLUS

Opinnäytetyö

TEKIJÄ/T: Simo Venäläinen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Simo Venäläinen	
Työn nimi PATU – SYÖPÄTUTKIMUSSOVELLUS	
Päiväys 25.4.2014	Sivumäärä/Liitteet 39
Ohjaaja(t) Keijo Kuosmanen - lehtori, Arto Mannermaa - apulaisprofessori, sairaalageneetikko	
Toimeksiantaja/Yhteistyökumppani(t) Itä-Suomen yliopisto	
<p>Tiivistelmä</p> <p>Tämä työ tehtiin Itä-Suomen yliopiston kliinisen patologian ja oikeuslääketieteen oppiaineelle suunnittelemalla ja toteuttamalla tietokantaa käyttävä sovellus, jolla tallennetaan syöpätutkimuksen tietoja sekä tuloksia. Sovellus toimii yhdellä tai useammalla tietokoneella ja tietokanta erillisellä palvelinkoneella. Työssä käytetty kanta oli testaamaton ja sitä piti vielä muokata muuttuneiden tarpeiden vuoksi.</p> <p>Työhön liittyi käyttöliittymän suunnittelu ja toteutus sekä tietokantayhteyden luominen sovelluksen ja tietokannan välille, jotka tehtiin käyttämällä Visual Studio 2012:sta. Lisäksi työssä käsiteltiin erilaisia lähestymistapoja toteuttaa sovellus. Toteuttamiseen oli valittavissa useampia erilaisia vaihtoehtoja. Tällaisia vaihtoehtoja olivat mm. C# Windows -sovelluskehys, ASP.NET web -ohjelmistokehys ja HTML/PHP -selainpohjainen sovellus.</p> <p>Valinnaksi tuli Windows-pohjainen C#:lla toteutettava WinForm, jonka avulla rakennettiin yksinkertainen mutta tehokas käyttöliittymä. Windows-sovelluskehukseen päädyttiin, koska WinForm soveltuu graafisten käyttöliittymien ja tietokantayhteyksien luomiseen graafisella suunnittelulla ja toteutuksella. Suunnittelu ja toteutus tapahtuivat yhtäaikaan, eli varsinaista suunnitteluvaihetta ei ollut. Työn kesto oli noin puoli vuotta.</p> <p>Tuloksena projektista syntyivät toimiva käyttöliittymäsovellus, tietokantasuunnitelmasta tehty toimiva MySQL-tietokanta sekä tämä raportti. Kaikki projektiin liittyvä materiaali luovutettiin asiakkaalle.</p>	
Avainsanat käyttöliittymäsovellus, syöpä	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Simo Venäläinen			
Title of Thesis Application for Cancer Research Database			
Date	25.4.2014	Pages/Appendices	39
Supervisor(s) Mr. Keijo Kuosmanen - Lecturer, Mr. Arto Mannermaa, PhD, Adjunct Professor			
Client Organisation /Partners University of Eastern Finland / Mr. Arto Mannermaa, PhD, Adjunct Professor			
<p>Abstract</p> <p>This project was carried out for The University of Eastern Finland, Institute of Clinical Medicine, Pathology and Forensic Medicine. The purpose of this project was to construct an application for existing cancer research database. The Application works on one or multiple computers while the database is installed in a separate server.</p> <p>The database was untested and needed some modifications because of the changed requirements. The Interface design and implementation was also part of the project. Additionally, this project included a comparison of different approaches to carry out the application.</p> <p>There were many different ways to create this application. These included a C# windows application framework, ASP.NET web software framework and HTML/PHP web-based application. The choice was the Visual Studio 2012 development environment with C# WinForm to create a simple but yet powerful interface. Visual Studio 2012 was chosen because it provides a graphical user interface and database links.</p> <p>The project was a very challenging experience. As a result, a long expected application to store cancer research data and a working MySQL database was constructed. All these were delivered to the client.</p>			
Keywords Interface application, cancer			

SISÄLTÖ

1	JOHDANTO	5
2	KÄYTETYT TERMIT JA LYHENTEET	6
3	SYÖVÄN MÄÄRITTELY.....	7
3.1	Solu	7
3.2	Syöpäsolu.....	8
4	SOVELLUKSEN MAHDOLLISET TOTEUTUSTAVAT JA NIIDEN VERTAILU	9
4.1	Selainpohjainen sovellus.....	9
4.2	Windows-sovellus	10
4.3	Web-ohjelmistokehys	11
4.4	Ohjelmiston valinta	13
5	TIETOKANTA	14
6	KÄYTTÖLIITTYMÄ	15
6.1	Rakenne.....	15
6.2	Toiminnot.....	16
6.2.1	Kirjautuminen	16
6.2.2	Valintasivu	17
6.2.3	Uuden statuksen lisääminen	18
6.2.4	Näytteen lisääminen olemassa olevaan statukseen	21
6.2.5	Statuksen tietojen muokkaus	22
6.2.6	Tietojen massasyöttö tiedostosta	24
6.2.7	Hakusivu	26
7	OHJELMAN RAKENNE.....	33
7.1	dbAvaayhteys-luokka	34
7.2	KirjautuminenLuokka.....	35
7.3	Status ja näyteluokat	35
7.4	YhteysLuokka	36
7.5	HakuLuokka.....	36
7.6	TarkistusLuokka.....	36
7.7	CSVLuku.....	37
7.8	CSVTarkistus	37
8	YHTEENVETO.....	38

1 JOHDANTO

Tämän työn tilasi Itä-Suomen yliopiston klinisen lääketieteen laitoksen patologian ja oikeuslääketieteen oppiaine. Laitos toimii Itä-Suomen yliopiston Canthia-rakennuksessa Savilahdessa. Laitoksella tutkitaan erilaisia syöpiä. Tämän lisäksi laitos vastaa patologian koulutuksesta lääketieteen opiskelijoille sekä erikoislääkärikoulutuksesta. Tämä opinnäytetyö käsittelee käyttösovelluksen suunnittelemista ja toteuttamista tietokantaan, jonka on suunnitellut ja toteuttanut Henri Jokitörmä omassa opinnäytetyössään. (JOKITÖRMÄ, 2013)

Työ tehtiin ratkaisemaan ja toteuttamaan jo olemassaolevalle tietokantaratkaisulle toteutuskelpoinen sovellus. Sovelluksen avulla voidaan lisätä ja hakea tietoa. Syöpätutkimuksesta tuleva tiedonmäärä on niin suuri, että sen syöttö tietokantaan käsin olisi hyvin hankalaa. Jokainen tutkija tallentaa tiedon omiin erillisiin tiedostoihin, jotka säilytetään kunkin tutkijan omalla koneella. Tieto jakautuu useaan eri tiedostoon ja siksi tietokanta luotiin tätä ongelmaa ratkaisemaan.

Sovelluksen avulla tutkijat voivat tallentaa yhteen paikkaan kaikki tutkimustiedot. Lisäksi entisestä toimintatavasta ei tarvitse luopua, vaan se toimii uuden käyttösovelluksen ohella. Tutkimustieto tallennetaan edelleen tiedostoihin, joista ne sitten luetaan ohjelmallisesti tietokantaan. Muutamia tietoja kuitenkin syötetään käsin käyttöliittymän avulla. Näitä ovat esimerkiksi uuden näytteen mukana tulevat potilaan tiedot sekä näytteen sijaintipaikka pakastimissa. Tutkimustulosten ohella sovelluksen avulla näytteiden sijainnin tarkistaminen helpottui. Lisäksi tutkijat voivat projektin sisällä tutkia kollegan tuloksia tarvitsematta mennä hänen tietokoneelleen.

Lopputuloksena luotiin yksinkertainen mutta tehokas käyttöliittymä, jonka avulla tieto saatiin talteen myöhempää käyttöä varten tutkijoille. Käyttöliittymän avulla tieto tallennettiin yhteen paikkaan, jolloin tieto ei enää pirstaloitu samalla tavalla. Vanha tiedostopohjainen tallennustapa säilyi uuden sovelluksen ohella, koska se todettiin hyväksi suurten tietomäärien siirtokanavaksi tietokantaan.

2 KÄYTETYT TERMIT JA LYHENTEET

Visual Studio	- Microsoftin kehittämä ohjelmointityökalu
NetBeans	- Ohjelmointiympäristö useammalle kielelle. Javapohjaisena vaatii Javan ajoympäristön (JRE)
SQL	- Tietokantakyselykieli, minkä avulla suoritetaan toimintoja tietokannassa
Käyttöliittymä	- Rajapinta, jonka avulla sovelluksen käyttäjät kykenevät käyttämään sovellusta
PHP	- Web-palvelinympäristössä käytettävä ohjelmointikieli, jota käytetään HTML:n kanssa luotaessa dynaamisia internet-sivuja.
HTML	- Yleinen nettisivujen luomiseen käytetty hypertekstin merkintäkieli.
ASP.NET	- Microsoftin kehittämä ohjelmointikieli web-ohjelmistokehys, jota käytetään dynaamisten web-sivujen luomisessa.
CSS	- Tyyliohjeiden laji, joka on suunniteltu erityisesti www-pohjaisten sivujen tyylien määrittelyyn.
Java	- Laitteistoriippumaton ohjelmointikieli, joka pohjautuu olioihin. Hyödyntää virtuaalista ajoa sekä luokkakirjastoja.
WinForm	- Windows Form. Graafinen sovellusohjelmointikäyttöliittymä windows-sovelluskehitykseen.
C#	- Microsoft:n .NET Framework:lle kehittämä ohjelmointikieli, joka yhdistää C++:n tehokkuuden ja Javan helppokäyttöisyyden.
Luokka	Olion tyyppi. Luokka määrittelee siitä luodun olion rakenteen ja käyttäytymisen. Kyseessä on siis eräänlainen olion rakennusmuotti.
Olio	Sovellusta ajettaessa luokasta luodaan olio, joka käyttäytyy luokan määrittelemällä tavalla.
CSV	Yksinkertainen taulukkotietoa sisältävä tekstitiedosto. CSV on siis toteutukseltaan tekstitiedosto, missä kentät on eroteltu toisistaan pilkuilla ja rivinvaihdolla. Tätä tiedostomuotoa voidaan lukea taulukkolaskentaohjelmalla.

3 SYÖVÄN MÄÄRITTELY

Syöpätutkimuksessa tutkitaan erilaisia syöpiä, ja niiden kehittymistä. Rintasyöpä on kuitenkin asiakkaan tämän hetkinen yleisin tutkimuskohde. Syöpäsolut ovat hyvin suuressa osassa tässä tutkimuksessa. Sen vuoksi seuraavissa kahdessa osiossa käydään hieman taustaa siitä, mitä solut ja syöpäsolut ovat ja miten ne käyttäytyvät. Näin saadaan hieman lisäymmärrystä yliopiston syöpätutkimuksen taustalla olevasta tärkeästä tutkittavasta materiaalista, joka vaikuttaa jokapäiväisessä elämässä meitä kaikkia.

3.1 Solu

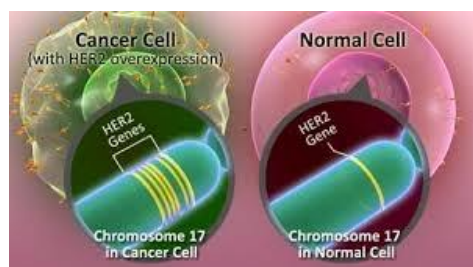
Ihmiset koostuvat pienistä silmälle näkymättömistä soluista. Ne sisältävät perintötekijämme, jotka määräävät mitä me olemme. Solut jakautuvat säännelysti riippuen niiden sijainnista elimistössä. Iho ja limakalvot ovat herkkiä vaurioitumaan, joten siellä solujen jakautuminen on vilkasta läpi ihmiselämän. Aivoissa solut kehittyvät ja jakautuvat, kunnes ihminen saavuttaa aikuisuuden. Tällöin puhutaan noin kahdestakymmenestä yhdestä ikävuodesta. Tähän mennessä aivojen solut lakkaavat jakautumasta. Pitkälle erilaistuneiden solujen, kuten aivon solujen, kohdalla saavutetaan siis piste, jolloin ne eivät enää jakaudu, kun taas esimerkiksi ihon tai luuston solut jakautuvat erilaistumisprosessin kautta tarvittaessa. Ikä vaikuttaa kaikkeen solujen jakautumiseen heikentävästi. (Syöpäjärjestöt, 2014-01-09)

Kudosvaurioiden kohdalla erilaistumisprosessi käynnistää valmiiksi kasvaneiden solujen jakautumisen, joka jatkuu kunnes vauriot on korjattu. Solujen määrä pysyy vakiona aikuisella terveellä ihmisellä. Solut muodostavat kudoksia. Kudostyyppi vaikuttaa siihen, millainen syöpä on kyseessä. Elimistössä on useita erilaisia kudostyyppisiä kuten epiteeli, tukikudos, hermokudos, imukudos sekä luuydin. Epiteeli tarkoittaa päällyskerroksia iholla ja limakalvoilla ja kudoksissa. Sen tehtävä on suojata, erittää ja imeä. Tukikudoksia ovat luu, rusto, sidekudos ja lihaskudos. Hermokudoksia löytyy etupäässä aivoista mutta hermosoluja pitkin etenevät kaikki aivoihin ja sieltä pois päin kulkevat impulssit kaikkialla kehossa. Imukudos estää mikrobien pääsyn elimistöön. Tämä kudos on tärkeimpiä elimistön immuunipuolustuksen osia. Sidekudokset toimivat siirtotienä kudosten välissä. (Syöpäjärjestöt, 2014-01-09)

Solun sisällä vaikuttavat solun kasvua kiihdyttävästi onkoproteiinit ja solua hidastavana tuumorisuppressorit. Näiden avulla solu kasvaa tasaisesti ja hallitusti. Onkoproteiineja ja tuumorisuppressoreita tuotetaan normaaleissa soluissa tasapainoisesti, mutta syöpäkudoksessa näitä proteiinien tuotantoa sekoittavat mutaatiot, jolloin solun säätelyjärjestelmä häiriintyy vakavasti. Tätä syöpäsolussa tapahtuvaa kasvua valaisee Turun yliopiston internetsivujen artikkelissa haastateltu Jukka Westermarck. Westermarck toteaa, että onkoproteiini on kuin kaasua. Se saa auton liikkumaan, tai tässä tapauksessa syövän kasvuun, vain jos kaasua painetaan. Jos tuumorisuppressor on aktiivisena, eli jarru painettuna alas, ei tapahdu syövän kasvua. Myöskään, jos kumpikaan pedaaleja ei paineta, ei pääse syntymään syövän hallitsemattomaa lisääntymistä. Tämä autovertaus on hyvä ja osuva kuvailemaan solun sisällä tapahtuvaa tasapainottelua hallitun kasvun ja hallitsemattoman syöpään johtavan kasvun veitsenterällä. (HYYTIÄINEN, Erja, 2014-01-07).

3.2 Syöpäsolu

Terve solu voi vaurioitua niin pahasti, että se muuttuu pahanlaatuisiksi. Tällöin puhutaan termistä karsinogeneesi, eli solun vaiheittainen muuttuminen pahanlaatuisiksi. Solussa olevan geenistön kasvua rajoittava signaalijärjestelmä vaimenee, tai jopa sammuu kokonaan. Tällöin kasvua kiihdyttävät signaalit voimistuvat aiheuttaen solun hallitsemattoman jakautumisen. Jakautumiseen vaikuttaa solun ympäristö. Tässä ympäristössä solujen, tiettyjen kasvua kiihdyttävien aineiden, hormonien ja hermoston signaalit vaikuttavat solun geenien toimintaan. Solun muututtua syöpäsoluksi se alkaa monistamaan itseään luoden pahanlaatuisia syöpäsolujoukkoja, kasvaimia, jotka suurenevat ja levittävät pahimmillaan etäpesäkkeitä muualle kehoon. Mutatoituneen solun jakautuessa uudestaan ja uudestaan isoksi joukoksi syöpäsoluja syntyy kyhmy, jota kutsutaan kasvaimeksi. Solujen tulee kuitenkin jakautua hyvin moneen kertaan, jopa tuhansia kertoja, että se olisi yhtä iso kuin herne. (Mikä on syöpä, 16-05-2013).



Kuva 1. Syöpäsolu ja terve solu (Chica-loc 2014-01-09)

Kuva 1 selventää normaalin, terveen solun ja syöpäsolun eroavaisuutta. Kromosomissa 17 sijaitseva HER2-geeni on mutatoitunut syöpäsolussa ja poikkeaa huomattavasti terveessä solussa olevasta.

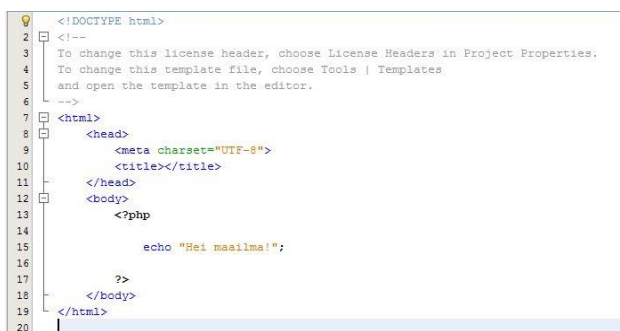
4 SOVELLUKSEN MAHDOLLISET TOTEUTUSTAVAT JA NIIDEN VERTAILU

Sovelluksen suunnittelun alkuvaiheessa tuli esille kysymys millä ohjelmointityökalulla ja tekniikoilla projekti toteutetaan. Valinnan varaa oli runsaasti ja päätös oli aluksi vaikea, kunnes raamit ohjelmistolle alkoivat hahmottua. Vaihtoehtoina oli selain- , Windows- ja web-ohjelmistopohjainen sovellus. Seuraavissa kappaleissa käydään läpi näiden eri vaihtoehtojen hyvät ja huonot puolet sekä mihin ratkaisuun päädyttiin ja miksi.

4.1 Selainpohjainen sovellus

Selainpohjaista sovellusta käytetään verkkoselaimen avulla. Sovellusta voidaan käyttää verkon kautta fyysisesti vaikka toiselta puolen maapalloa, eli sen käyttö ei ole sidottu yhteen paikkaan. Tällainen ratkaisu on hyvä, jos toiminta on liikkuvaa ja tietoihin tulisi päästä käsiksi käytännössä missä vain. Yleensä tällainen verkkosovellus yhdistetään toimimaan palvelimella olevan tietokannan kanssa. Tällöin sovelluksessa tuotettu tieto saadaan tallennettua myöhempää käyttöä varten. Selainpohjaisten sovellusten kohdalla tietoturva on hyvin tärkeä pitää mielessä, sillä niiden pohjana olevissa ohjelmointikielissä piilee huonosti toteutettuna riski tietoturvan heikentymisestä.

Selaimella käytettävien sovellusten tekemiseen on monia hyviä työkaluja. Ohjelmointikielinä tähän tarkoitukseen on käytössä esimerkiksi JAVA ja PHP-kielet. PHP on kielenä yleiskäyttöinen, joka soveltuu hyvin web-pohjaisten sovellusten kehitykseen, kuten dynaamisten sivustojen luomiseen. Lisenssivapaina PHP on hyvä valinta, jos halutaan välttää kehitysvaiheessa ongelma kaupallisten oikeuksien kanssa. Kuka tahansa voi siis käyttää maksutta tätä kieltä ohjelmointitarkoituksiin. Kieli on siis helppo oppia ja käyttää, mutta samalla sen hankaluus on tietoturvassa, sillä PHP on altis tietomurroille tietokantakyselyjä tehdessä sekä tiedonsiirrossa sivulta sivulle. JAVA on virtuaaliympäristössä ajettava alustariippumaton oliopohjainen kieli, jolla voidaan luoda selaimessa ajettavia sovelmia, dynaamisia www-sivuja. JAVA:ssa käskyt kulkevat virtuaalikoneen läpi, jolloin varmistetaan etteivät käskyt tule ulkopuolelta. Näiden rajoitusten vuoksi JAVA:lla tehdyt ohjelmat on muita turvallisempia, mutta hieman hitaampia.



```

1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8 <head>
9 <meta charset="UTF-8">
10 <title></title>
11 </head>
12 <body>
13 <?php
14     echo "Hei maailma!";
15
16 <?>
17 </body>
18 </html>
19
20

```

Kuva 2. Esimerkkikuva PHP:n perussyntaksista

Kuva 2 osoittaa perusrakenteen PHP-ohjelmoinnissa. Kuvan mukaisten PHP-tagien sisään sijoitettava PHP-koodi voidaan sijoittaa HTML:n sekaan. Nämä tagit sisältävät toiminnallisuuden, joka PHP:llä halutaan toteuttaa. Sovelluksen www-sivuilla käyttäjä ei näe missään vaiheessa tätä koodia, sillä se toimii vain palvelimella. PHP ajetaan palvelimella PHP-tulkin avulla. Tämän esiprosessin aikana tulkki kääntää ja suorittaa koodin. Lopputuloksena tulkki palauttaa HTML-muotoista koodia asiakkaan näyttöpäätteelle. Tietokantapohjainen SQL-kyselykieli toimii hyvin PHP:n kanssa, sillä PHP:n avulla voidaan SQL-komentoja toteuttaa kätevästi tietokantaan. CSS-tyylitiedostot menevät oivallisesti HTML:n ja PHP:n kanssa selainohjelmoinnissa määritellen sivuille niiden ulkoasun.

Selainpohjaiset sovellukset ovat yleensä kokoelma internetsivuja ja erilaisia kooditiedostoja. Tällaisten ohjelmien tekemiseen löytyy esimerkiksi ohjelmointityökalu Netbeans. Se on ilmainen ja soveltuu HTML/PHP-ohjelmointiin sekä Java-pohjaisten ohjelmien tekemiseen. Myös Microsoftin Visual Studio ohjelmointityökalu sopii myös mutta maksullisena mutta lisenssien vuoksi se ei välttämättä ole sopiva valinta kaikille.

4.2 Windows-sovellus

Windows-ympäristössä voidaan ohjelmoida konsolipohjaisia sovelluksia ja WinFormia (API), jonka avulla voidaan luoda sovelluksia, jotka hyödyntävät graafisia lomakkeita. Näiden lomakkeiden avulla WinForm-sovellusta voidaan käyttää. WinFormeja ohjelmoitaessa käytetään C#-ohjelmointikieltä. Lomakkeet ovat pääosassa WinFormissa.



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Poista
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

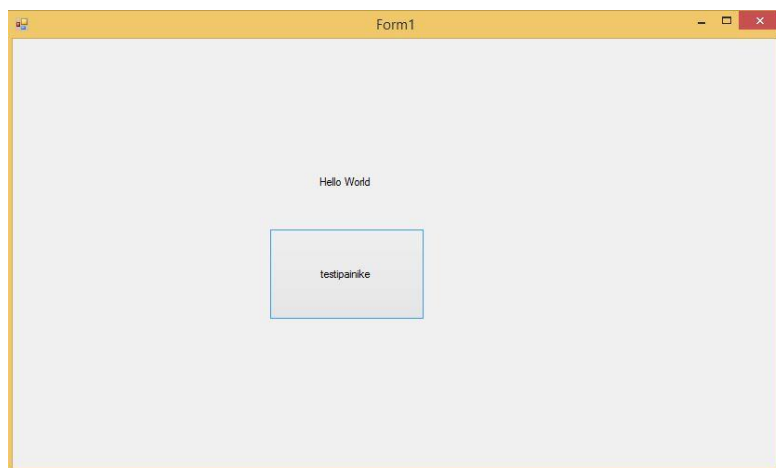
        private void TestiFunktio()
        {
            label1.Text = "Hello World";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            TestiFunktio();
        }
    }
}

```

Kuva 3. C#:n koodiosion esimerkki

Kuva 4 näyttää pelkistetyn esimerkin lomakkeesta. Yleensä sovellusten lomakkeet voivat olla hyvinkin monimutkaisia ja syvällisempiä. Oikealla sijoittelulla saadaan kuitenkin monimutkaisinkin lomake käyttäjäystävälliseksi.



Kuva 4. C#:n lomake-esimerkki

Kuva 3 näyttää yksinkertaisen Hello World esimerkin taustakoodin. Kuva 4 ilmaisee samaisen koodin toiminnon lomakkeella, missä on painettu painiketta. Tämän jälkeen label1:n teksti on muuttunut.

Microsoft kehittää myös Windows-käyttöjärjestelmiä. Windowsille ohjelmia luotaessa voidaan käyttää erilaisia työkaluja. Näistä parhaimpien joukkoon kuuluu Microsoftin Visual Studio. Tämän ohjelmiston huono puoli on sen maksullisuus ja lisenssit. Visual Studion hyviin puoliin kuuluvat kätevä graafinen käyttöliittymä, monipuoliset työkalut ja tietokantayhteysmahdollisuudet.

Ilmaisia työkaluja löytyy myös. Aiemmin mainitulla Netbeans-ohjelmistolla olisi mahdollista ohjelmoida Windowsille konsolipohjaisia sovelluksia. Netbeans kuitenkin soveltuu parhaiten verkkosivujen ohjelmointiin, sekä Java-pohjaisten ohjelmien tekemiseen.

4.3 Web-ohjelmistokehys

Microsoft tarjoaa dynaamisten verkkosivujen, verkkopalvelujen ja ohjelmien ohjelmointiin oman ohjelmistokehityksen. ASP.NET toimii .NET-kielellä, eli C#:lla. ASP.NET:n ohjelmointi tapahtuu Visual Studion kautta, joten se on periaatteessa maksullinen ohjelmointikieli. Ilmaisia kehitysympäristöjä on olemassa, mutta Visual Studio on ylivoimaisesti paras valinta, sillä Visual sisältää kätevät graafiset työkalut erilaiseen ohjelmoimiseen sekä tietokantayhteyksiin.

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Poista.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<p><asp:Label ID="Label1" runat="server" Text="Label"></asp:Label></p>
<p><asp:Button ID="Button1" runat="server" Text="Button" /></p>
</div>
</form>
</body>
</html>

```

Kuva 5. ASP.NET:n aspx-koodiosio

Kuva 5 näkyy ASP.NET:n HTML ja aspx-koodi. Tämän kautta on mahdollista luoda sovelluksen käyttöliittymä ja muuta toimintoa, kuten tietokantayhteyksiä. Käyttöliittymän luominen onnistuu myös graafisena, mutta tämä on hieman hankalampaa. Koodin käyttö on kaikkein helpointa.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Poista
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        private void TestausFunktio()
        {
            Label1.Text = "Hello World";
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            TestausFunktio();
        }
    }
}
```

Kuva 6. ASP.NET C# taustakoodi



Kuva 7. ASP.NET Esimerkkisovellus selaimessa

Kuva 6 nähdään sovelluksen taustalla oleva koodi. ASP.NET:ssä ohjelmoidaan myös C#:lla tietyin eroavaisuuksin. C#:n hallinnasta on siis hyötyä ohjelmoitaessa ASP.NET:llä. Kuva 7 ilmentää, miltä kuvien 5 ja 6 koodi näyttää selaimessa. ASP.NET:llä ja WinForm:lla on tietyt valintakriteerit sovelluksiin liittyen. Jos halutaan luoda isoja verkkosovelluskokonaisuuksia, on ASP.NET oikea valinta. Pienempiä sovelluksia varten soveltuu WinForm. Näin on tilanteissa, joissa verkko ei ole itseisarvo sovelluksen käyttämiselle, eli sovellus toimii myös ilman verkon yli tapahtuvaa liikennettä.

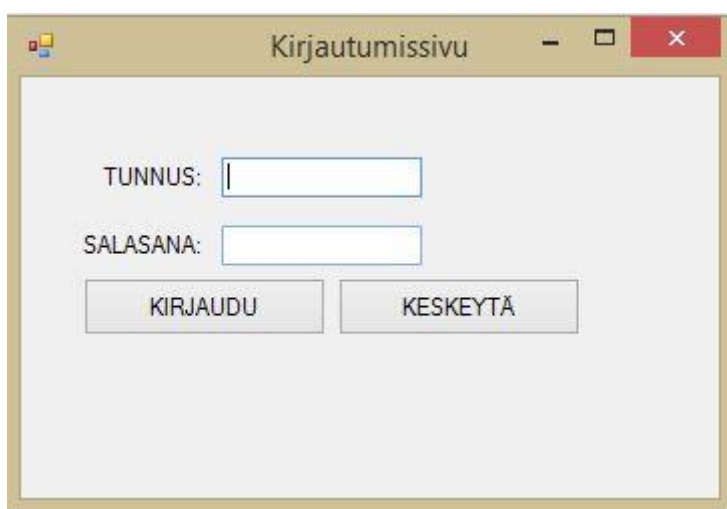
4.4 Ohjelmiston valinta

Loppujen lopuksi valinta ohjelmistokielen ja työkalun valinnassa ei ollut kovinkaan vaikea. Tähän vaikutti yliopistolta löytynyt lisenssi Visual Studion käyttämiseksi sekä se, että C#:lla ohjelmoimisesta löytyi vahvaa osaamista. Visual Studiossa on hyvät tietokantayhteysmahdollisuudet ja WinForm-ohjelmointi mahdollistaa kätevien sovelluslomakkeiden luomisen. ASP.NET olisi muuten tullut kyseeseen, mutta tarvetta internetin hyödyntämiseen selaimen muodossa ohjelmiston käytössä ei ollut. Sovellusta on siis tarkoitus käyttää paikallisesti tutkimustiloissa. Neljän vuoden opiskeluaikana Visual Studiota on käytetty hyvin paljon, eli tuttu ohjelma helpottaa työtä jatkossakin.

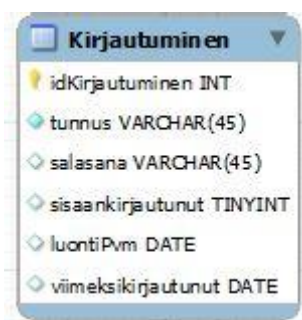
6.2 Toiminnot

6.2.1 Kirjautuminen

Sovelluksen käyttäminen lähtee aina kirjautumisesta. Sovellusta käyttävät tutkijat syöttävät omat henkilökohtaiset tunnukset lomakkeelle. Tunnukset tarkistetaan tietokannassa olevasta taulusta, missä kirjautumistiedot sijaitsevat. Kuva 10 esittää sovelluksen kirjautumislomakkeen. Kuva 11 esitellään tietokannassa oleva taulu, joka sisältää kirjautumistiedot. Kirjautumisessa otetaan talteen sekä tunnuksen luontipäivämäärä, että milloin käyttäjä on viimeksi ollut kirjautuneena. Potilaiden tietoturva tulee näin suojattua. Taulusta löytyvää sisäänkirjautumisen ilmaisevaa kenttää ei sovelluksessa käytetä ja se on lisättyä mahdollista myöhempää tarvetta varten.



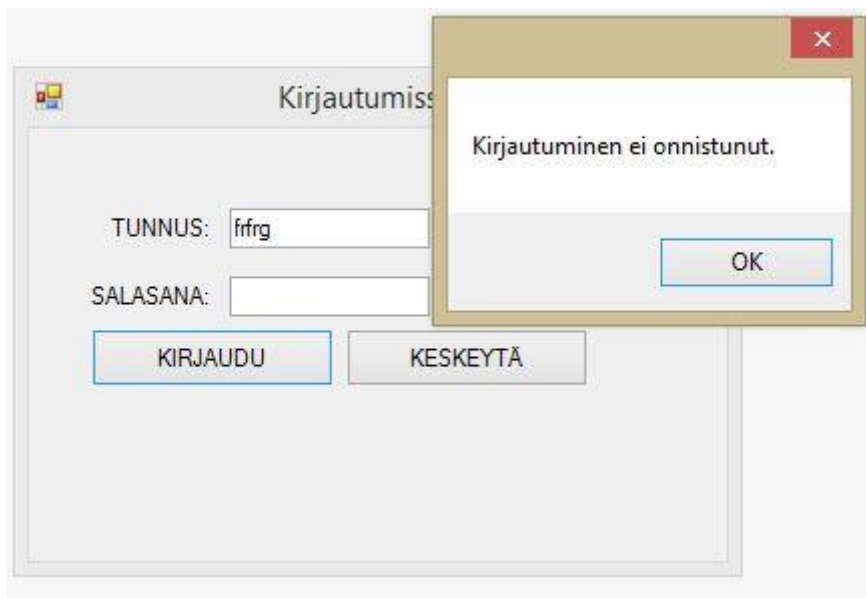
Kuva 10. Kirjautuminen



Kirjautuminen	
idKirjautuminen	INT
tunnus	VARCHAR(45)
salasana	VARCHAR(45)
sisaankirjautunut	TINYINT
luontiPvm	DATE
viimeksikirjautunut	DATE

Kuva 11. Kirjautumistaulu

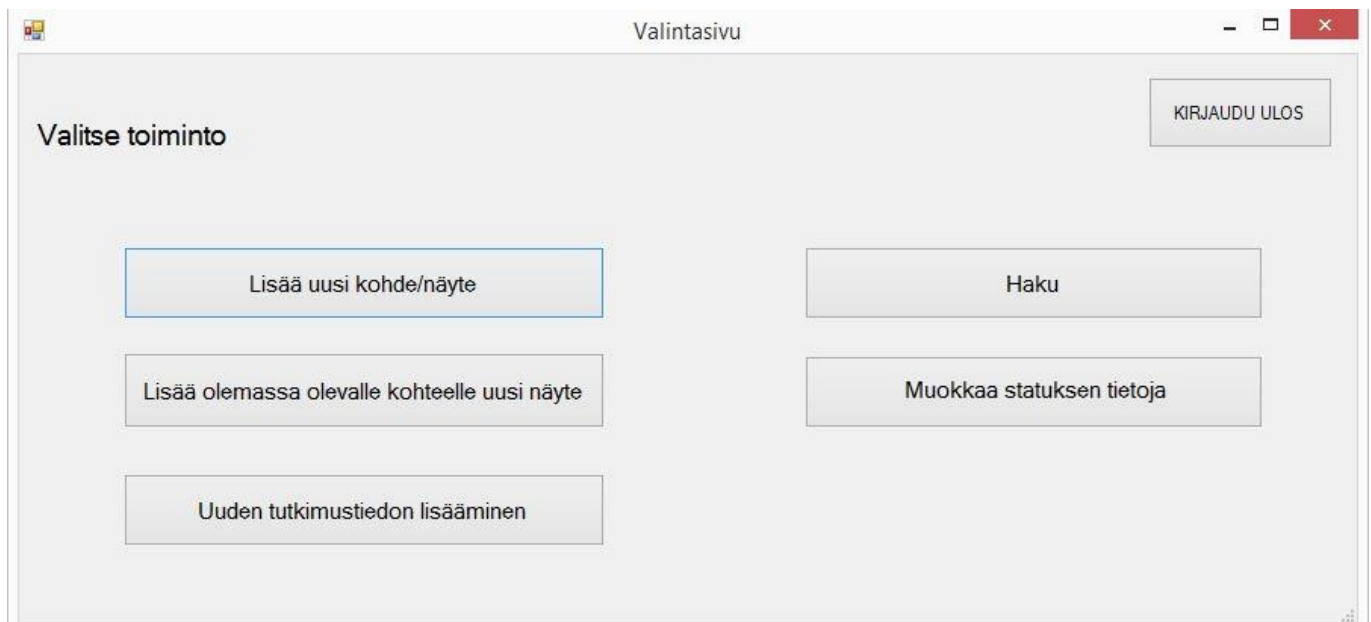
Väärän tunnuksen ja/tai salasanan kohdalla käyttäjälle annetaan virheilmoitus, joka on esitetty Kuva 12. Epäonnistuneita syöttökertoja ei ole rajoitettu, eli sisäänkirjautumista voi yrittää niin monta kertaa kuin haluaa. Tästä ei ole haittaa, sillä sovellusta käytetään vain tutkimustiloissa, joihin on pääsy vain tutkijoilla. Toki tutkijoidenkin joukossa on sellaisia, jotka eivät liity juuri sovelluksen käyttötarkoituksen piiriin kuuluvaan tutkimustyöhön.



Kuva 12. Epäonnistunut kirjautumisyritys

6.2.2 Valintasivu

Valintasivulta löytyvät kaikki sovelluksen toimintopainikkeet, sekä uloskirjautumispainike, joka kirjaa käyttäjän ulos ja sulkee sovelluksen. Painikkeet on nimetty selkeästi, jotta käyttäjälle olisi mahdollisimman selkeää, mitä kukin painike tekee. Valintasivulla on vain painikkeet varsinaisiin toimintoihin. Paneeli on pyritty pitämään mahdollisimman yksinkertaisena, sillä täällä suoritetaan vain sovelluksen toimintojen valinta. Valintasivu esitellään Kuva 13.



Kuva 13. Valintasivu

6.2.3 Uuden statuksen lisääminen

Tietokantaan lisätään tämän lomakkeen avulla uuden statuksen perustiedot. Lomake koostuu päivämäärän syöttöosiosta, missä annetaan haastattelu ja diagnoosipäivämäärät alavetovalikoista, statuksen perustietoosiosta ja näyteosiosta. Molemmat päivämäärät voidaan jättää halutessaan syöttämättä, koska aina ei kyseisiä tietoja ole saatavilla heti. Tällöin voidaan kenttien alla olevista valintaruuduista estää päivämäärien syöttö. Kuva 14 esittää lomakkeen rakennetta. Statuksen tyyppi valitaan alavetovalikosta. Valikossa on valittavissa potilas, verrokki ja solulinja.

Solulinja poikkeaa muista siten, että sille ei anneta muita tietoja kuin nimi. Potilaan ja verrokin tiedoille kuuluvat kentät menevät harmaaksi, kuten Kuva 16 nähdään. Jokainen status liittyy johonkin projektiin, joten siihen on oma syöttökenttensä. Henkilötunnus annetaan muodossa pppkvv-xxxx. Tätä tietoa käytetään varmentamaan olemassa olevan statuksen olemassaolo ja tulevan näytteen liittyminen haluttuun statukseen. Käyttäjälle tulostetaan tunnuksen etuosa varmennusta varten.

Potilasnumero on tärkeä, koska se liittää potilaan, verrokin ja solulinjan tiedot statukseen. Statuksen potilasnumero ja näytteen potilasnumerokentät ovat liitoksessa toisiinsa, koska molempiin tulee sama tieto. Näyteosiossa on potilasnumerolle kenttä, jotta näyte voidaan helpommin liittää statukseen. Näin eliminoidaan myös kirjoitusvirheiden mahdollisuus. Näytetietoja ei voi sivuuttaa, koska näyte tulee aina tallentaa tietokantaan. Muuten tulee ongelmia mihin näytteeseen tutkimustiedot liittyvät, kun tietoja ei voida liittää otettuun näytteeseen. Tietokantaan ei tule yhtäkään statusta ilman näytettä, sillä näyteputkilon kyljessä on statuksen tiedot merkittynä. Statuksen ollessa tietokannassa voidaan siihen lisätä uusia näytteitä erillisellä lomakkeella. Verrokkia, eli terveen vertailuhenkilön tietoa ei ole tarpeen syöttää, jos sitä ei ole. Silloin -999 syötetään verrokin kenttään. Cambridgenumero on tärkeä, sillä näytteitä lähetettäessä muualle täytyy potilasnumero koodata uusiksi.

Näytteelle annetaan tiedot kuten tyyppi, putkinumero, potilasnumero, näytteen koko, kappalemäärä ja päivämäärä, jolloin kyseinen näyte on otettu. Nämä tulevat huolimatta statuksen tyyppivalinnasta poislukien kudosnäytteet, joiden kohdalla kokotietoa ei tarvita. Kuva 15 näyttää näytteiden tyypin valintalistan, mistä valitaan näytteelle sen tyyppi. Näytteen sijaintitiedot annetaan siten, että tutkija käy viemässä näytteen vapaaseen paikkaan ja syöttää sitten tiedon lomakkeelle. Yksi näyteputkilo jaetaan useammaksi näytteeksi kappalemäärän perusteella. Tämä toteutetaan koodissa silmukassa, jonka kierrosmäärän edellä mainittu kenttä määrittää. Jokaisella silmukan lisäämällä näytteellä on sama näytenumero, sillä ne ovat samaa näytettä jaettuna useampaan putkiloon. Käytetty näyte merkitään nollaamalla näytteen koko. Ei ole väliä minkä putkilon koon muuttaa. Tietokantaan jää tieto tyhjästä putkilosta.

Uusi Status
tässä ilmoitus->

STATUS: Tyyppi: -Valitse status-

Haastattelu:
Päivä: 20
Kuukausi: 3
Vuosi: 2014

☐ Syötetään myöhemmin

Diagnoosipäivämäärä:
Päivä: 20
Kuukausi: 3
Vuosi: 2014

☐ Syötetään myöhemmin

Etunimi:
Sukunimi:
Synnyinkunta:
Kotikunta:
Syntymäpäivä: 20 3 2014
Henkilötunnus:
Kenen verrokki:
Potilasnumero:
Cambridgenro:

Projekti:
Solulinjan nimi:

(Jos ei verrokkia, niin syötä -999
voi jättää myös tyhjäksi)

NÄYTE:
Tyyppi: -Valitse näytetyyppi-
Näytenumero:
Potilasnumero:
Koko: ml.
Määrä: Kpl.
Näytteenottopvm: 20. maaliskuuta 2014

SIJAINTI:
Huone:
Pakastin:
Lokero:
Räkki:
Laatikko:

Paluu
valintasilulle

Tallenna tiedot

Kuva 14. Uuden statuksen lisäyslomake

Tietoja syötettäessä sovellus varmistaa, että tärkeimmät tiedot tulevat annetuksi, kuten statuksen ja näytteen tyyppi. Sovellus antaa Kuva 17 mukaisen virheilmoituksen, jos tallennuspainikkeen painamisen jälkeen havaitaan puutteita lomakkeen tiedoissa. Samalla kyseinen kenttä korostetaan punaisella lukuunottamatta alasvetovalikoita. Virheen sattuessa tiedot eivät katoa kentistä luoden käyttäjäystävällisempää toimintamallia sovellukseen. Tietojen tallentamisen jälkeen lomakkeen kentät tyhjennetään valmiiksi mahdollista seuraavaa statuksen lisäystä varten.

-Valitse näytetyyppi-

- Kudosnäyte
- Iso seerumi
- Pieni seerumi
- Parafiininäyte
- Hepariniveri
- EDTA-veri
- RNA
- RNA(veri)
- RNA(tuorekudos,tuumori,primääri)
- RNA(tuorekudos,tuumori,metastaasi)
- RNA(parafiini,tuumori,primääri)
- RNA(parafiini,tuumori,metastaasi)
- DNA
- DNA(veri)
- DNA(tuorekudos,tuumori,primääri)
- DNA(tuorekudos,tuumori,metastaasi)
- DNA(parafiini,tuumori,primääri)
- DNA(parafiini,tuumori,metastaasi)
- Proteiini
- Solunappi
- Elävät solut
- Muu näyte

Kuva 15. Näytteiden valintalista

Uusi Status
tässä ilmoitus->

STATUS: Typpi: Solulinja

Haastattelu:
 Päivä: 20
 Kuukausi: 3
 Vuosi: 2014
☐ Syötetään myöhemmin

Diagnoosipäivämäärä:
 Päivä: 20
 Kuukausi: 3
 Vuosi: 2014
☐ Syötetään myöhemmin

Henkilötiedot:
 Etunimi:
 Sukunimi:
 Synnyinkunta:
 Kotikunta:
 Syntymäpäivä: 20 3 2014
 Henkilötunnus:
 Kenen verrokki:
 Potilasnumero:
 Cambridgenro:

Projekti:
Solulinjan nimi:

(Jos ei verrokkia, niin syötä -999
voi jättää myös tyhjäksi)

NÄYTE:
☐ Syötetään myöhemmin
 Typpi: -Valitse näytetyyppi-
 Näytenumero:
 Potilasnumero:
 Koko: ml.
 Määrä: Kpl.
 Näytteenotto: 20. maaliskuuta 2014

SIJAINTI:
 Huone:
 Pakastin:
 Lokero:
 Räkki:
 Laatikko:

Kuva 16. Uusi statuslomake solulinja valittuna

Henkilötunnus:
 Kenen verrokki:
 Potilasnumero:
 Cambridgenro:

NÄYTE:

Virhepotilasnumero
 Anna potilasnumero

kkia, niin syötä -999
myös tyhjäksi)

Kuva 17. Virhe potilasnumeron syötössä

6.2.4 Näytteen lisääminen olemassa olevaan statukseen

Tietokannassa olevaan statukseen voidaan lisätä uusia näytteitä kätevästi tämän lomakkeen avulla. Potilasnumeron avulla haetaan tietokannasta sitä vastaava numero, mihin näyte lisätään. Ennen varsinaista tietokantalisäystä tarkistetaan, että potilasnumero on oikea. Potilasnumero voi silti löytyä kannasta ja tällöin näyte saattaa pahimmillaan kirjautua aivan väärään statukseen. Tutkija saa näytölle ilmoituksen syötetyn potilasnumeron perusteella joko löydetyn potilaan syntymäaika tai solulinjan nimen. Näin voidaan tarkistaa näytteen mukana tulleista näytetiedoista, että lomakkeelle syötetty potilasnumero on oikein kirjoitettu ja sen löytämä potilastieto täsmää näytteen mukana tulleiden tietojen kanssa. Samalla varmistetaan, että potilaasta otettua näytettä ei liitetä solulinjan viljelmään.

Näytteen Lisäys olemassa olevaan statukseen

Potilasnumero:

Näytteenottopvm: 20. maaliskuuta 2014

NÄYTTEEN TYYPI:

SIJAINTI:

Huone:

Pakastin:

Pakastinlokero:

Rakki:

Laatikko:

NÄYTE:

Näyttenumero:

Koko: ml.

Määrä: Kpl.

PARAFIINIINÄYTE:

Näyttenumero:

koko: ml.

Määrä: Kpl.

Kuva 18. Olemassa olevaan statukseen näytteen lisäyslomake

Lomakkeella syötetään myös näytteenottopäivämäärä, tyyppi ja sijainti sekä putkinumero ja näytteen koko ja kappalemäärä. Sijaintitiedot syötetään samalla tavalla, kuin uuden statuksen lisäyslomakkeella. Näyteosio muuttuu harmaaksi ja parafiininäyte aktivoituu, kun valitaan tyyppiä tuorenäyte. Näytteelle ja parafiininäytteelle on omat syötekentät, sillä molemmat tallennetaan tietokantaan erillisiin tauluihin. Kuva 18 havainnollistaa, miten näyte-osio

on valmiiksi valittuna ja parafiininäyte harmaana. Parafiininäytetyyppi valittuna osiot vaihtavat tilaa. Tästä on esimerkkinä Kuva 19. Tiedot tarkistetaan myös tällä lomakkeella. Tyhjät kentät aiheuttavat virheilmoituksen ja kehotuksen virheen korjaamiseksi. Kappalemäärä määrittää moneen erään näyte jaetaan, kuten uuden statuksen lisäyksessäkin.

Näytteen Lisäys olemassa olevaan statukseen

Potilasnumero:

Näytteenottopvm: 20. maaliskuuta 2014

NÄYTTEEN TYYPI: DNA(parafiini,tuumori,metastaasi)

Paluu valintasivulle

Tallenna tiedot

SIJAINTI:

Huone:

Pakastin:

Pakastinlokero:

Rakki:

Laatikko:

NÄYTE:

Näyttenumero:

Koko: ml.

Määrä: kpl.

PARAFIININÄYTE:

Näyttenumero:

koko: ml.

Määrä: kpl.

Kuva 19. Parafiinityyppi valittuna

6.2.5 Statuksen tietojen muokkaus

Statuksen tietoja voidaan muokata lisäyksen jälkeen, sillä haastattelu ja diagnoosipäivämääriä ei aina ole mahdollista syöttää statuksen tietoihin. Muitakin aiemmin esitellyillä lomakkeilla syötettyjä tietoja voidaan muokata. Tällaisia tietoja ovat tällä hetkellä potilaan nimitiedot, verrokkietieto sekä näytteet.

Muokkauslomakkeella on aluksi käytettävissä vain potilasnumeron haku, kuten Kuva 20 nähdään. Haettava tieto tarkistetaan ja, jos kyseessä on solulinja, ilmoitetaan haetun solulinjan nimi. Potilaan kohdalla vastaavasti näytetään syntymäpäivä eli henkilöturvätunnuksen alkuosa. Halutun potilasnumeron syötön jälkeen kaikki kentät aktivoituvat ja niihin ilmestyy vastaavat potilasnumeron tiedot. Kuva 21 havainnollistaa muokkaustilan aktivoinnista seuraavat muutokset. Lisäksi haetaan potilasnumeroon liitetyt kaikki näytetiedot viereiselle listalle. Näin sovelluksen käyttäjä voi samalla halutessaan tarkistaa näytteiden määrän ja päivittää

näytteiden käyttöastetta. Nämä tiedot tulevat sekä potilailta, että solulinjoilta. Solulinjan kohdalla vain näytelistä aktivoituu ja näyttää haetut tiedot.

Kuva 20. Muokkauslomake

	putkinro	tyyppi	statusnro	koko
▶	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
	muu_002	Muu näyte	3	1
*				

Kuva 21. Muokkauslomake muokausvalmiudessa

Haastattelu- ja diagnoosipäivämäärät voidaan lisätä varsinaisen statuksen lisäyksen jälkeen tai korjata mahdollisesti aiemmin tulleet syöttövirheet. Muokkauslomaketta on mahdollista kehittää lisää miettimällä siihen mahdollisesti uusia muokattavien tietojen kenttiä, kuten solulinjan nimi ja kuntatiedot.

6.2.6 Tietojen massasyöttö tiedostosta

Tutkijat ovat tähän asti tallentaneet tutkimustietonsa CSV-tiedostoihin. CSV-tiedosto on Excel-taulukon tapainen tekstitiedostomuoto. Tietoja voi olla hyvin paljon, joiden lisääminen käsin tietokantaan voi muodostua hankalaksi, ellei jopa mahdottomaksi. Käsinsyöttö lisää myös virheiden todennäköisyyttä. Tähän ongelmaan tuli löytää ratkaisu. Tietojen keskittäminen kokonaisuuksittain yksittäisiin tiedostoihin tarjosi yhden ratkaisun. Tällä tarkoitetaan esimerkiksi immunologisten tietojen keskittämistä yhteen CSV-tiedostoon. Tätä periaatetta käyttäen kaikki tietokokonaisuudet koottuna omiin erillisiin tiedostoihin tuo helpotusta ohjelmalliseen tietojen lukemiseen.

Tämä osa sovelluksesta oli muita osioita hankalampi toteuttaa. Erityisesti tietojen turvallinen syöttäminen tietokantaan oli vähintäänkin vaikea toimenpide. Jopa tuhansien sarakkeiden lukeminen tietokantaan ei ole yksinkertainen toteuttaa. Syväsekvointi-taulut jouduttiin jättämään rauhaan ja odottamaan jatkokehitystä, koska CSV-tiedostot vaativat niiden kohdalla sen verran paljon aikaa miettiä mitä tietoja niihin laitetaan, että projektin aika ei siihen olisi riittänyt. Lisäksi laboratoriossa pitää miettiä mitä tietoja kantaan laitetaan, joten tämä päätettiin jättää jatkokehittäväksi ominaisuudeksi. Kuva 22 havainnollistaa tietojen massalähettämiseen käytettävän lomakkeen rakennetta ja ulkoasua.

Kuva 22. Tietojen massalähettämislomake

Kun haluttu tiedosto on valittu Valitse tiedosto-painikkeen avulla, tulee ruutuun tumman harmaalle alueelle kyseisen tiedoston tiedot. Kyseisen tiedoston on oltava CSV-tiedosto, muussa tapauksessa ohjelma antaa virheilmoituksen ja estää vääränlaisen tiedoston syöttämisen. Kuva 23 esittää tilannetta, missä CSV-tiedosto Snippidata.csv on haettu ja aukaistu Valitse tiedosto -painikkeen avulla. Kyseisen tiedoston tiedot tulevat näkyville tumman harmaalle datagridview:n alueelle. Samalla näytetään esikatselun alalaidassa tiedostopolku, mistä kyseinen tiedosto on haettu. Lomakkeen vasemmalla reunassa olevasta syöttövalintapaneelista tulee valita minne tieto ollaan syöttämässä. Tässä vaiheessa käyttäjä voi tarkistaa tiedoston tiedot ja tarvittaessa palata vielä muokkaamaan tiedostoa. Muussa tapauksessa painetaan Lisää tiedot tietokantaan -painiketta. Tällöin käyttäjälle ilmoitetaan kuinka monta riviä ollaan lisäämässä. Toiminnan jälkeen tulee ilmoitus onnistuneesta tietojen lisääksestä.

pnrro	rs268all	rs764all	BCAC5_0TNRC9
34	1	1	12
42	1	1	12
43	1	1	22
44	2	1	11
46	1	1	12
*			

Kuva 23. CSV-tiedosto esikatselutilassa

Koodi syöttövalintojen takana muodostui melko lailla vastaavanlaiseksi jokaisen osa-alueen kohdalla. CSV-tiedostosta luetaan tiedot datatableen ja viedään sellaisena CSVLuku-luokan olio. Siellä kutsutaan syöttövalintaa vastaavaa metodia. Datatablesta tutkitaan sen rivien ja sarakkeiden lukumäärä. Uloimpana silmukkana pyöritään rivien mukaisesti rivi riviltä. Jokaisella rivillä käydään sisemmässä silmukassa sarake sarakkeelta jokainen kenttä, joista luetaan arvot tietokantaan. Esimerkiksi snippi-tietojen kohdalla luetaan snippien nimet otsikkoriviltä ja yhdistetään ne sitten vastaavalta sarakekentästä löytyvään arvoon. Missään vaiheessa käyttäjän ei tarvitse tietää mitä kenttien nimiä tai arvoja löytyy, sillä ohjelma lisää tarvittaessa nimet omiin tauluihinsa. Jos arvokentissä sattuu olemaan tyhjiä kohtia, ne on merkitty numerolla 777. Tästä ohjelma tietää, että kyseessä on arvoton kenttä, jolloin se ei lisää sitä tietokantaan. Esimerkkinä snippitauluissa tällaista arvotonta snippiä ei tarvitse lisätä. Jos myöhemmin kyseiselle snipille saadaan arvo, voidaan se lisätä uudessa lisäyksessä uusien snippitietojen kanssa.

Potilaiden taustatutkimus ja kliinisten tietojen lisääminen on myös melko vastaavanlainen snippien kanssa. Kyseisten taulujen tiedot CSV:ssä luetaan rivi ja sarake kerrallaan. Yhden rivin tiedot tulevat ikään kuin yksittäisinä osina, mutta ne sitoutuvat toisiinsa statusnumeron avulla.

6.2.7 Hakusivu

Sovelluksen tehokas käyttö edellyttää hakutoimintoja. Haulla voidaan hakea erilaisia perustietoja, kuten potilaan historiaan ja kliinisiin hoitotietoihin liittyviä tietoja. Hakusivulla on luotu osiot yleisimmille osa-alueille, joita ovat kliiniset tiedot, taustatiedot, immunologia, syväsekvenssi, dna ja rna. Näissä hakuehtona on yleisesti potilasnumero, jonka avulla haetaan status-tilusta statusnumero. Näytteitä voidaan hakea myös näytteen omalla yksilöivällä näyttenumerolla, jolloin vain kyseisen näytteen putkilot tulevat haussa haetuksi.

Muutosten vuoksi syväsekvenssi jäi kokonaan pois sovelluksen toteutuksesta ja sitä myöten myös hakutoiminnoista. Se toteutetaan myöhemmin erillisenä projektina. Tästä syystä kyseinen osa-alue jätettiin sovelluksen käyttöliittymiin valmiiksi tulevaa kehitystyötä varten. Projektin aikataulun vuoksi päätettiin keskittyä hakutoiminnoissa vapaaseen hakuun, eli käyttäjät voivat valita valmiista vaihtoehdoista itse hakuparametrit, joiden avulla haetaan tietoa. Yleisempiä hakutoimintoja kehitettiin sen verran, että syötetyn tiedon hakeminen tietokannasta saatiin testattua.

Kuva 24 esittää hakusivun, missä ovat kaikki valinnat ja toiminnot. Ylälaidasta löytyy yleiset hakutoiminnot. Lomakkeen keskellä on täsmennetty hakuosio. Käyttäjän syöttämillä hakuehdoilla on mahdollista hakea tietojoukkoja antamalla ennakkoon määriteltyihin valintakenttiin halutut ehdot. Tätä toimintoa on mahdollista kehittää edelleen lisäämällä toisenlaisia hakuehtoja, joilla haetaan muuta täsmennettyä tietoa tietokannasta. Tallenna CSV-tiedostoon –painikkeella on mahdollista tallentaa haetut tiedot erilliseen CSV-tiedostoon, jotta haettua tietoa olisi mahdollista hyödyntää muuallakin.

The screenshot shows a web application window titled "Haku". The interface includes a top navigation bar with tabs for "Potilas-tiedot", "DNA-tiedot", "RNA-tiedot", "Proteiinitiedot(immunologia)", and "Näytteet". The main content area features three search filter inputs: "Ikä:" (Age), "Geenimuunnos:" (Gene mutation), and "Proteiinin värjäys kasvaimesta:" (Protein staining in tumor). Below these inputs are two buttons: "Paluu valintasivulle" (Return to selection page) and "Hae ehtojen mukaisesti" (Search according to criteria). A large, empty grey box occupies the center-right of the page, intended for displaying search results. At the bottom right, there is a button labeled "Tallenna CSV-tiedostoon" (Save to CSV file).

Kuva 24. Hakutoimintojen pääsivu

Potilashaussa hakuehtoina käytetään potilasnumeroa. Sovellus hakee kyseiseen potilasnumeroon liittyvän statusnumeron ja hakee siihen liitetyt potilaan tiedot. Samalla lomakkeella voidaan hakea tiedot joko taustatutkimustauluista tai vastaavista kliinisten hoitotietojen tauluista. molemmissa osa-alueissa tiedot ovat kahdessa eri taulussa. CSV-tiedostosta tallennetaan nimitiedot omaan tauluun ja siihen liittyvä tieto omaan. Hakutulokset esitetään vastaavalla tavalla. Kuva 25 havainnollistaa lomakkeen sijoittelua. Valinta-osiossa voidaan päättää, kumpia tietoja halutaan hakea.

The screenshot shows a web application window with a title bar. Inside, there are two main sections at the top: 'Tiedot' (left) and 'Nimet' (right), each containing a large, empty rectangular box. Below these boxes, on the left, are two buttons: 'Paluu valintasivulle' and 'Hae potilaan tiedot'. To the right of these buttons is a text input field labeled 'Syötä potilasnumero:'. Further right is a section labeled 'valinta' containing two radio buttons: 'Kliiniset tiedot' and 'Taustatutkimustiedot'.

Kuva 25. Potilaan tietojen hakulomake

Kuva 26 näyttää miten potilasnumerolla on haettu kliinisiä tietoja tietokannasta. Valinta kliinisten hoitotietojen ja taustatutkimustietojen välillä tehdään valinta-osiossa. Arvot tulostuvat vasemmanpuoleiseen taulukkoon ja niiden nimet oikeanpuoleiseen. Syy tällaiseen järjestelyyn on siinä, että molemmat tiedot ovat eri tauluissa, joista ne on erikseen myös haettava. Kuva 27 on haettu vastaavasti potilasnumerolla taustatutkimustiedot. Samalla tavalla kuin kliinisten tietojen haussa myös taustatutkimustaulussa näkyvät arvot vasemmalla ja niiden nimet oikealla. Jatkokehitystä ajatellen tätä hakua voitaisiin kehittää siten, että nimet tulisivat haetuksi arvojen lukemisen yhteydessä ja näin voitaisiin syöttää kaikki tieto samaan taulukkoon.

Tiedot

	tieto	idhoito
▶	40	374
	kuollut, muu syy	375
	dead	376
	2/25/1993	377
	2,891170431	378
	no relapse (local ...	379
	2/25/1993	380
	2,891170431	381
	vasemman rinnan...	382
	on tehty kainaloe...	383

Nimet

	idtunnus	nimi
	374	TAM_ANNOS
	375	STATUS
	376	ALIVE_DEAD
	377	FU_DATE
	378	FU_SURVIVAL
	379	RFS
	380	RFS_DATE
	381	RFS_SURVIVAL
	382	OIK_VAS
	383	KAIN_EV

Paluu valintasivulle Hae potilaan tiedot

Syötä potilasnumero: 46

valinta

☒ Kliiniset tiedot

☐ Taustatutkimustiedot

Kuva 26. Potilaan kliinisten tietojen haku

Tiedot

	tieto	idtaustatiedon_nimi
	3431	187
	3431	188
	42	189
	ERITTAin RASK...	190
	160	191
	77	192
	12	193
	KYLLa	194
	56	195
	TA ELaVaNa SY...	196

Nimet

	idtunnus	nimi
	187	amm
	188	ntyo
	189	nvv
	190	tluo
	191	spit
	192	spai
	193	kkika
	194	rask
	195	rpv1
	196	rp1

Paluu valintasivulle Hae potilaan tiedot

Syötä potilasnumero: 34

valinta

☐ Kliiniset tiedot

☒ Taustatutkimustiedot

Kuva 27. Potilaan taustatietohaku


Näytteitä voidaan hakea sekä potilasnumerolla, että putkinumerolla. Kuva 28 osoittaa, kuinka näytteiden hakulomake asettaa potilasnumeron oletuksena hakuvaihtoehdoksi. Tätä voidaan vaihdella valinta-osion valintanapeilla. Haku tuo näytteistä näkyville tietoja kuten nimen, tyyppin, koon millilitroissa, näytteenottopäivämäärän ja tiedon missä näyte sijaitsee. Mahdolliset potilasnumeron statukseen lisätyt tuorenäytteet tulevat myös hakutulokseen mukaan. Niille tulee samat tiedot mutta poikkeuksena niiden tyyppi-kenttään ei tule mitään, koska niille ei anneta tyyppiä. Niiden tyyppi selviää tuorenäytteen nimestä ja/tai potilasnumerosta. Kuva 29 esittää onnistuneen näytteiden haun potilasnumerolla.

Kuva 28 Näytehaku

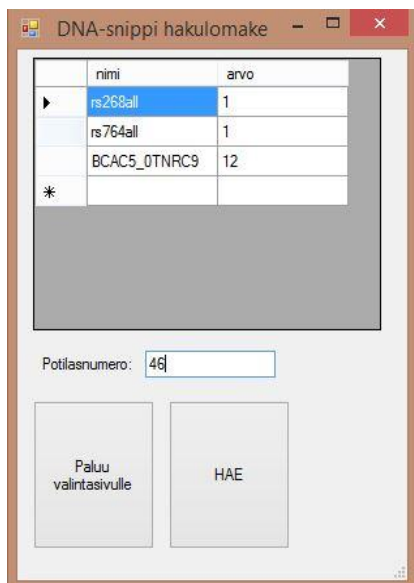
nro	tyyppi	koko	pvm	laatikko
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6
edta_567	EDTA-veri	1	2.4.2014	6

Kuva 29 Suoritettu näytteiden haku potilasnumerolla

DNA-snippien hakulomake Kuva 30 on hyvin pelkistetty johtuen siitä, että varsinaisia tietokenttiä haussa on vain pari. Ne ovat Snipin nimi ja sen arvo. Potilasnumerolla haetaan tässäkin hakuosiossa vastaava statusnumero, jonka avulla suoritetaan varsinainen tietojen haku. Kuva 31 nähdään kuinka potilasnumerolla 46 on saatu hakutuloksena kolmen eri snipin tiedot.



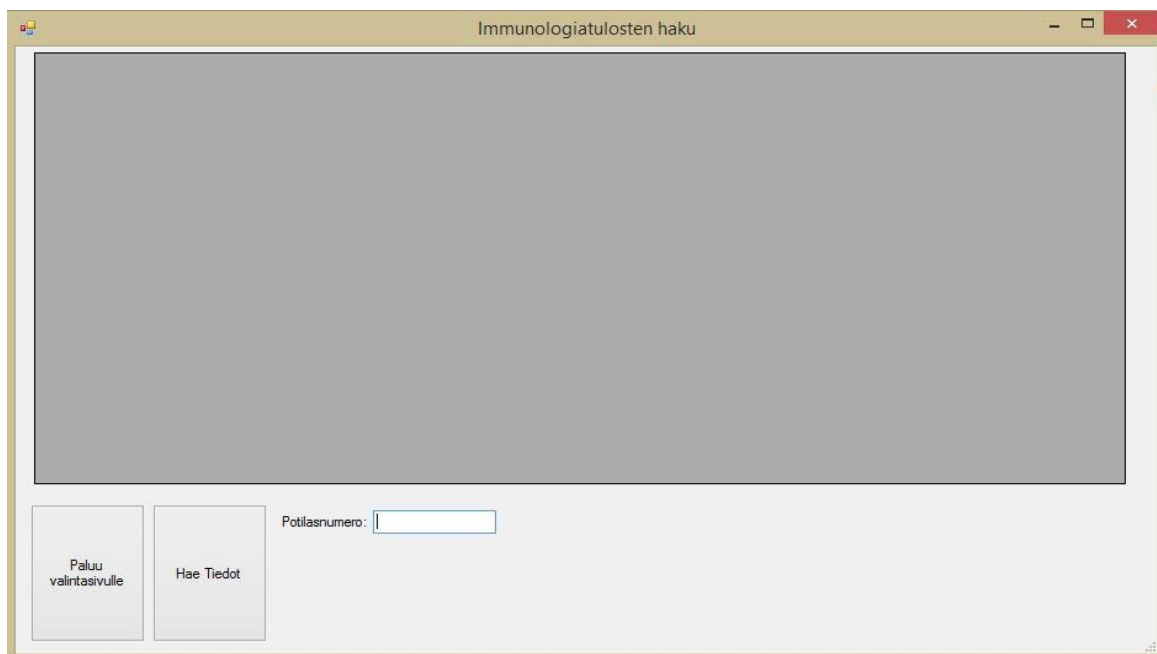
Kuva 30. DNA-snippien hakulomake



nimi	arvo
rs268a11	1
rs764a11	1
BCAC5_0TNRC9	12
*	

Kuva 31. DNA-snippien hakulomake hakutulos

Immunologisten tietojen haku ei eroa muista hakutoiminnoista oleellisesti. Kuva 32 ilmenee immunologisten tietojen hakulomakkeen ulkoasu. Potilasnumerolla haetaan myös täällä tietoja tietokannasta. Kuva 33 on haettu potilasnumerolla tietoja ja ne on nähtävillä näytöllä.



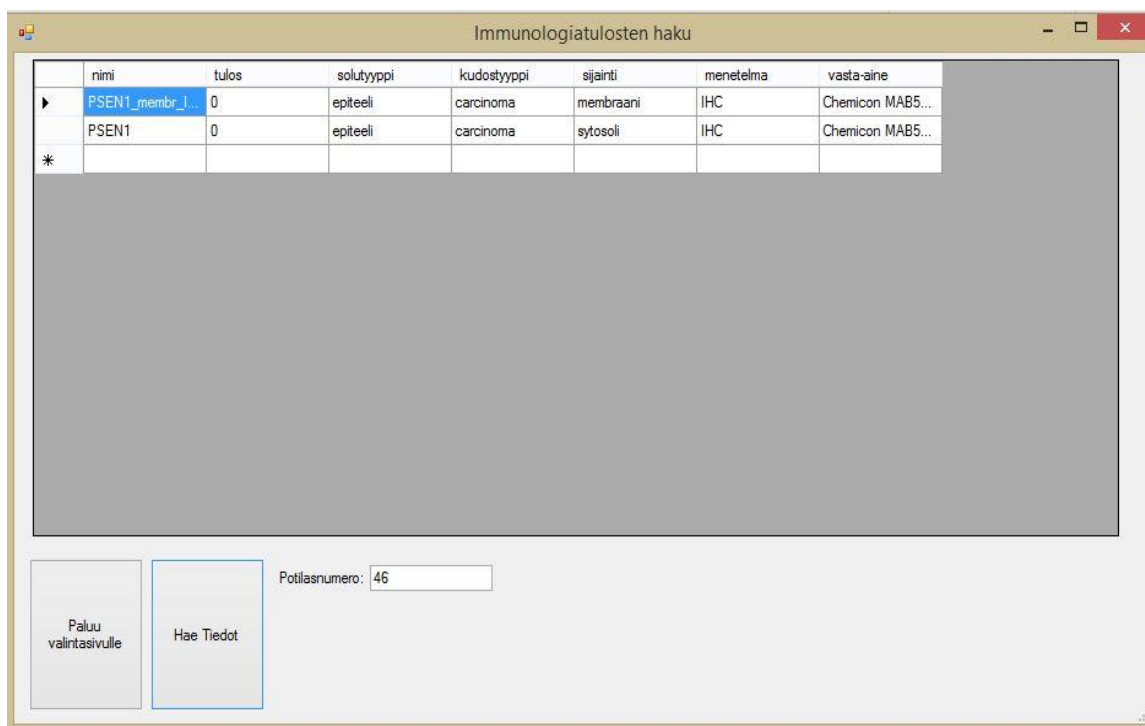
Immunologiatulosten haku

Paluu valintasivulle

Hae Tiedot

Potilasnumero:

Kuva 32. Immunologisten tietojen hakulomake



Immunologiatulosten haku

	nimi	tulos	solutyyppi	kudostyyppi	sijainti	menetelma	vasta-aine
▶	PSEN1_membr_I...	0	epiteeli	carcinoma	membraani	IHC	Chemicon MAB5...
	PSEN1	0	epiteeli	carcinoma	sytosoli	IHC	Chemicon MAB5...
*							

Paluu valintasivulle

Hae Tiedot

Potilasnumero: 46

Kuva 33. Immunologisten tietojen haku suoritettu

Rna-ekspressioiden haku toteuttaa samaa kaavaa muiden hakujen kanssa. Tiedot haetaan potilasnumeron perusteella Kuva 34 lomakkeelle. Kuva 35 nähdään hakutulokset, kun potilasnumerolla on haettu tietokannasta siihen liitettyt rna-tiedot. Jos potilasnumeroa ei syötä, ei tapahdu mitään eikä siitä erikseen ilmoiteta.

Kuva 34. Rna-hakulomake

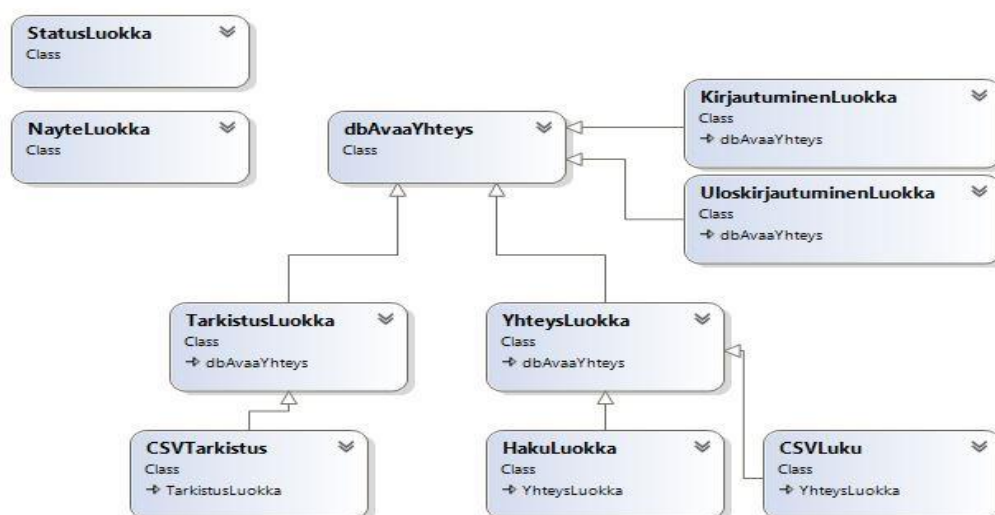
	geeni	tulos	kontrolli	assay	menetelma
▶	PEN2	0,545695364	PPIA	Hs01033961_g1	Taqman
	PSEN1	0,172753623	PPIA	Hs00997789_m1	Taqman
	PSEN1	0,172753623	PPIA	Hs00997789_m1	Taqman
	PSEN2	0,594269341	PPIA	Hs01577197_m1	Taqman
*					

Kuva 35. Rna-haku suoritettu

7 OHJELMAN RAKENNE

Sovellus käyttää erilaisia luokkia toimintojensa suorittamiseen. Sovellukseen syötetyt tiedot liikkuvat olioissa ja siirretään tietokantaan ja tietokannasta käyttöliittymään olioiden välityksellä. Lomakkeiden taustakoodissa tarkistetaan tiedot ja tämän jälkeen kutsutaan get-metodeja. Tieto kulkee muuttujiin suoraan. StatusLuokan ja NäyteLuokan lisäksi sovellus hyödyntää monia muita luokkia. Näihin lukeutuvat mm. tarkistus, -yhteys, -haku- ja kirjautumislukut. Näistä luokista on enemmän alempana.

Luokat jakautuvat siten, että osa luokista periyttää ominaisuuksia toiselta luokalta. Kuva 36 näyttää luokkien väliset suhteet. dbAvaaYhteys-luokkaa tarvitaan käytännössä kaikissa luokissa, paitsi näytteiden ja statustietojen luokissa. Uloskirjautuminen-luokkaa ei käytetä missään, sillä ominaisuus mihin se olisi tarpeellinen ei ole vielä käytössä. Käytännössä kyseinen luokka on valmiina mahdollista jatkokehitystä silmällä pitäen sovelluksen kirjautumistoimintoihin. Tällä hetkellä kirjautuminen tietokantaan jättää tiedon vain viimeisimmästä kirjautumisesta, mutta varsinaista paikallaolotietoa ei tallenneta. Tietokannan kirjautumistaulusta voidaan nähdä, että kyseinen ominaisuus löytyy, mutta sille ei toistaiseksi ole tarvetta.



Kuva 36. Luokkakaavio

7.1 dbAvaayhteys-luokka

DbAvaayhteys on luokka, joka hoitaa yhteyden luomisen tietokantaan. Samalla se myös vastaa yhteyden alustamisesta. Kuva 37 näyttää merkkijonon, jonka avulla luodaan yhteys tietokantaan. Samasta kuvasta näkee myös MySqlCommand-tyyppisen muuttujan cmd. Cmd on käytössä jokaisessa tietokantakyselyssä. Tämän vuoksi se täytyy alustaa jokaisen kyselyn jälkeen. Tarkistuksen hoitaa k

Kuva 39 havainnollistaa tarkistuskoodi. DbAvaayhteys-luokkaa kutsutaan ylivoimaisesti eniten sovelluksessa johtuen sen luonteesta yhteyksien luomisen hallinnassa.

```
string MyconnectionString = "Server=localhost;Database=patu;Uid=root;Pwd=root;";
public MySqlCommand cmd = new MySqlCommand();
MySqlConnection conn;
```

Kuva 37. Kirjautumismerkkijono

MySqlConnection-tyyppinen muuttuja conn ottaa parametrina MyconnectionString-muuttujan sisältämän merkkijonon Kuva 38 mukaisesti. Conn:ia käytetään vain varsinaisen tietokantayhteyden luomiseen. Cmd saa CreateCommand()-metodilla conn:lta yhteyden kantaan, jolloin sitä voidaan käyttää kyselyiden toteuttamisessa. Tämä on esitetty Kuva 40. Koodi on try-catch-lohkon sisällä, jotta mahdolliset yhteysongelmat saadaan selville ja näkyville.

```
conn = new MySqlConnection(MyconnectionString);
```

Kuva 38. Tietokantayhdistämissmerkkijonon liitos kyselymuuttujaan

```
// Tarkistetaan, että conn-muuttuja on varmasti suljettu. Näin "alustetaan" yhteys kantaan
if (conn != null && conn.State == ConnectionState.Open)
{
    conn.Close();
}
```

Kuva 39. Yhteydentarkistus

```
// Aukaistaan yhteys ja asetetaan cmd-muuttuja käytettäväksi yhteydessä
conn.Open();
cmd = conn.CreateCommand();
```

Kuva 40. Yhteyden aukaisu ja cmd:n käyttöönotto

7.2 KirjautuminenLuokka

Tämän luokan ainoa tehtävä on tarkistaa tietokannan kirjautuminen-taulusta, että sieltä löytyy käyttäjän antamat tunnus ja salasana. Luokkaa hyödynnetään KirjautuminenUserControl:ssa, missä tarkistetaan KirjautuminenLuokan olion palauttaman Boolean arvo. Jos se on true, kirjautuminen hyväksytään. Muussa tapauksessa käyttäjälle ilmoitetaan virheellisestä kirjautumisyrityksestä. Kuva 41 voidaan nähdä kyseinen tarkistus. Olio nimeltä kirjautuminen kutsuu KirjautuSovellukseen()-metodia, jossa tarkistus suoritetaan. Ensin varmistetaan tunnuksen olemassaolo. Tunnuksen löytyessä jatketaan salasanan tarkistuksella. Metodi palauttaa arvon true, kun se löytää toisiaan vastaavat tunnus- ja salasanaparit. Muussa tapauksessa palautetaan arvo false.

```
if (kirjautuminen.KirjautuSovellukseen() == true)
{
    if (this.LoginOK != null)
        LoginOK(this, new EventArgs());
}
else
{
    if (this.LoginFail != null)
        LoginFail(this, new EventArgs());
}
```

Kuva 41. UserControl-tarkistus

7.3 Status ja näyteluokat

Käyttöliittymästä statukseen liittyvät tiedot tallennetaan StatusLuokasta luotuihin olioisiin. Kuva 42 esittää tämän luokan rakenteen. Vastaavasti näytetiedot menevät NäyteLuokan olioisiin. Tämän luokan rakenne näkyy Kuva 43. Kuvista nähtävät get ja set -metodit mahdollistavat sen, että erillisiä haku- ja syöttömetodeja ei luokkaan tarvitse luoda. Muuttujien ollessa julkisia ne näkyvät muualla sovelluksessa ja niitä voidaan näin kutsua suoraan. Näin tietojen kuljetus paikasta toiseen tapahtuu luontevammin ja huomattavasti tehokkaammin, kuin yksitellen siirtämisessä.

```
class StatusLuokka
{
    // Muuttujat

    // Status-taulun tiedot
    public string m_potilasno { get; set; }
    public string m_cambridge { get; set; }
    public int m_verrokki { get; set; }
    public string m_projekti { get; set; }
    public int m_kohde_id { get; set; }
    //Syntymäpäivä
    public int m_sp { get; set; }
    public int m_sk { get; set; }
    public int m_sv { get; set; }

    // haastattelupäivämäärä
    public int m_hpv { get; set; }
    public int m_hkk { get; set; }
    public string m_hvv { get; set; }
    // diagnoosipäivämäärä
    public int m_dpv { get; set; }
    public int m_dkk { get; set; }
    public string m_dvv { get; set; }

    // Kohde-taulujen tiedot
    public string m_tyyppi { get; set; }
    public string m_solulinja_nimi { get; set; }
    public string m_skunta { get; set; }
    public string m_nkunta { get; set; }
    public string m_etunimi { get; set; }
    public string m_snimi { get; set; }
    public string m_henk_tunnus { get; set; }
}
```

Kuva 42. StatusLuokka

```

class NayteLuokka
{
    // Muuttujat

    public string m_tyyppi { get; set; }
    public string m_putkinumero { get; set; }
    public string m_potilasnumero { get; set; }
    public int m_naytekoko { get; set; }
    public int m_kpl { get; set; }
    public string m_nayte_otto_pvm { get; set; }

    // Sijainti

    public string m_huone { get; set; }
    public string m_pakastin { get; set; }
    public string m_lokero { get; set; }
    public string m_rakki { get; set; }
    public string m_laatikko { get; set; }

    // Näytteen sijoitus varastoon
    public int laatikko_id { get; set; }
    public int haettu_laatikko_id { get; set; }
    public int rakki_id { get; set; }
    public int pakastinlokero_id { get; set; }
    public int pakastin_id { get; set; }
    public int huone_id { get; set; }
}

```

Kuva 43. NäyteLuokka

7.4 YhteysLuokka

Yhteysluokka sisältää kaikki ne metodit, joita tarvitaan tietojen siirtoon tietokantaan. YhteysLuokka hyödyntää tarkistusluokkaa jokaisessa metodissaan, sillä tietojen syöttäminen tietokantaan ilman asianmukaisia tarkistuksia aiheuttaa ongelmia. Uuden statuksen lisäämisessä tulee tarkistaa, ettei annettua potilasnumeroa ole jo syötetty. Kunta-taulusta tarkistetaan, löytyykö taulusta syötetty kunta. Tällaisin erilaisin tarkistuksin YhteysLuokka lisää tietokantaan sieltä puuttuvia tarpeellisia tietoja ennen varsinaisen tiedon lisäystä.

7.5 HakuLuokka

YhteysLuokasta on periytetty HakuLuokka, joka myös suorittaa kyselyitä tietokantaan. DbAvaaYhteys-luokka periytyy YhteysLuokan kautta myös tälle luokalle, jonka ainoa tehtävä on hakea tietoa kannasta. HakuLuokkaa käytetään hakutoiminnoissa hakemaan esimerkiksi potilasnumerolla siihen liittyvää aineistoa tietokannasta. Tiedon näyttämisessä käyttäjälle ymmärrettävässä muodossa vastaa DataTable-tyyppiset muuttujat. Näitä muuttujia voidaan muokata siten, että niiden sarake ja rivitietoja voidaan lisätä ja nimetä tarpeen mukaan. DataTable sidotaan lopuksi DataView-tyyppiseen muuttujaan, joka taas yhdistetään DataSourceena datagridview:iin, joka on objekti lomakkeella.

7.6 TarkistusLuokka

Tämän luokan metodeja hyödynnetään etupäässä YhteysLuokassa, sillä tietojen syötössä täytyy tietokannasta löytyä vaaditut tiedot kuten vierasavaimiin tulevat viitearvot. Luokka sisältää joukon erilaisia metodeja, joiden avulla voidaan kätevästi tarkistaa tietokannan tietoja ja varmistaa niiden olemassaolo ja erityisesti puute. Ainoat luokat, joita TarkistusLuokka hyödyntää dbAvaaYhteys-luokan lisäksi, ovat StatusLuokka ja NäyteLuokka. Näiden kahden luokan olioiden mukana liikkuvat kaikki tarkistukseen vaaditut tiedot metodeihin.

7.7 CSVLuku

CSVLuku on YhteysLuokasta periytetty lapsiluokka ja sen tehtävään kuuluu CSV-tiedostojen lukeminen. Tämä luokka sisältää kaikki metodit, joiden avulla luetaan CSV-tiedostojen sisältämä tieto tietokantaan. CSVLuku-luokka hyödyntää CSVTarkistus- ja HakuLuokkaa sekä myös dbAvaayhteys-luokkaa. CSVLuku ei muista luokista poiketen käytä ollenkaan StatusLuokkaa eikä NäyteLuokkaa.

7.8 CSVTarkistus

CSVTarkistus-luokka on Tarkistus-luokan lapsiluokka. CSVTarkistus hoitaa tarkistukset CSV-tiedostojen lukukoodissa sisältäen kaikki ne tarkistusmetodit, joita tarvitaan kyseisessä toiminnassa. Myös CSVTarkistus-luokka ei käytä StatusLuokkaa eikä NäyteLuokkaa.

8 YHTEENVETO

Tavoitteena oli suunnitella ja luoda toimiva käyttöliittymäsovellus Itä-Suomen yliopiston kliinisen lääketieteen laitoksen patologian ja oikeuslääketieteen oppiaineelle tietoja varastoivan tietokannan päälle. Tietokanta aiheutti ongelmia, joiden vuoksi sitä täytyi päivittää muuttuneiden vaatimusten vuoksi. Käyttöliittymän suunnittelussa suureksi avuksi oli rakentava palaute tutkijoilta. Niiden pohjalta oli hyvä luoda sovellus, jonka käyttö olisi mahdollisimman helppoa ja vaivatonta.

Suurten tietomäärien tallentaminen tietokantaan aiheutti suurimman vaikeuden projektin aikana. Tiedon lukeminen CSV-tiedostoista oli haastavaa, sillä tästä ei projektia ennen ollut aiempaa kokemusta. Työn tekeminen oli hyvin avartavaa ja luovaa, ja myös haasteellista. Tiedon etsiminen oli haastavaa ja vaati paljon työtä. Patologian laitoksen tutkijoiden palaute auttoi erittäin paljon sovelluksen toiminnallisen koodin tuottamisessa sekä luonnollisesti käyttöliittymän luomisessa. Toimiva kommunikointi asiakkaan kanssa auttoi huomattavasti projektin etenemistä koko prosessin ajan.

Projektin aikana opin hyvin paljon, miten ohjelmat kannattaa suunnitella ja jäsentää koodi ymmärrettävällä ja selkeällä tavalla. Samalla ymmärrys luokkien hyödyntämisestä sovelluskehityksessä kehittyi huomattavasti. Luokat eivät ole enää pelkkiä yksittäisiä osia, vaan sovelluksen sisällä rakenteita yhdistäviä ja niiden välisen tiedonsiirron mahdollistavia rakennuspalikoita.

Projekti jäi sellaiseen vaiheeseen, että sitä on hyvä lähteä jatkokehittämään. Erityisesti syväsekvenssitietojen jättäminen pois luo kehitystarpeen, johon on hyvä tarttua myöhemmin. Myös hakutoimintoja olisi hyvä viedä eteenpäin ja kehittää monimuotoisemmaksi, jolloin tietokannan sisältämien tietojen hyödyntämisaste kasvaa.

LÄHTEET JA TUOTETUT AINEISTOT

Chica-loc. Minä ja sairas elämäni. Anarkisti 2013-12-01. [viitattu 2014-01-09].

Saatavilla: <http://kohdustahautaan.blogspot.fi/2013/12/anarkisti.html>

HYITIÄINEN, Erja. Miksi syöpäsolut selviävät paremmin kuin normaalisolut? [WWW-artikkeli].

[viitattu 2014-01-07]. Saatavilla: <http://www.utu.fi/fi/Ajankohtaista/Artikkelit/Sivut/miksi-syopasolut-selviavat-paremmiin-kuin-normalisolut.aspx>

JOKITÖRMÄ, Henri 2013. PaTu : Patologian tutkimustietokannan suunnittelu [viitattu 2013-12-27].

Saatavissa: <http://www.theseus.fi/handle/10024/58716>

LEHTINEN, Päivi. 2014. Syöpäsolu murtaa järjestyneen kudusrakenteen oman "sorkkarautansa" avulla. [WWW-dokumentti]. [viitattu 09.01.2014].

Saatavilla: <http://www.biomedicum.fi/index.php?page=1619&lang=1>

Syöpäjärjestöt. Ei päiväystä. Kudostyypit. [WWW-dokumentti]. [viitattu 09.01.2014].

Saatavissa: <http://www.cancer.fi/tietoasyovasta/syopa/kudostyypit/>

Syöpäjärjestöt. Ei päiväystä. *Mikä on syöpä*. [WWW-dokumentti]. [viitattu 28.12.2013].

Saatavissa: <http://www.cancer.fi/tietoasyovasta/syopa/>